

WHAT IS CLAIMED IS:

1 1. A method, comprising:
2 determining a number of conflicting entries in a first redirection table having a
3 first set of entries, wherein the first set of entries is capable of being mapped to a second
4 set of entries of a second redirection table; and
5 mapping the first set of entries to the second set of entries, based on the number of
6 conflicting entries in the first redirection table.

1 2. The method of claim 1, wherein the first redirection table is a software
2 redirection table, wherein the second redirection table is a hardware redirection table, and
3 wherein a conflict is caused if at least two entries of the software redirection table that are
4 capable of being mapped to one entry of the hardware redirection table indicate different
5 receive queues, the method further comprising:
6 determining whether the first set of entries in the software redirection table has
7 more members than the second set of entries in the hardware redirection table, wherein
8 the number of conflicting entries are determined in response to determining that the first
9 set of entries in the software redirection table has more members than the second set of
10 entries in the hardware redirection table; and
11 indicating that packets associated with conflicting entries are to be directed to one
12 receive queue, in response to determining that the number of conflicting entries is less
13 than a threshold.

1 3. The method of claim 2, further comprising:
2 distributing packets in the one receive queue among all processors for processing;
3 and
4 processing packets in other receive queues in different processors.

1 4. The method of claim 2, further comprising:

2 indicating that all packets are to be directed to a single receive queue, in response
3 to determining that the number of conflicting entries is not less than the threshold.

1 5. The method of claim 4, further comprising:
2 processing receive side scaling in software, wherein processing receive side
3 scaling further comprises creating virtual queues and queuing deferred procedure calls to
4 corresponding processors via a device driver.

1 6. The method of claim 2, further comprising:
2 programming the hardware redirection table in accordance with the software
3 redirection table, in response to determining that the first set of entries in the software
4 redirection table does not have more members than the second set of entries in the
5 hardware redirection table.

1 7. The method of claim 1, wherein determining and mapping are performed
2 by a device driver in a computational platform having a plurality of processors.

1 8. The method of claim 1, wherein the first redirection table is associated
2 with an operating system that supports receive side scaling, wherein the second
3 redirection table is implemented in a hardware device coupled to a computational
4 platform having a plurality of processors, and wherein the second redirection table is of a
5 fixed size.

1 9. A system, comprising:
2 at least one processor;
3 a network interface coupled to the at least one processor; and
4 program logic including code that is capable of causing the at least one processor
5 to be operable to:

6 (i) determine a number of conflicting entries in a first redirection table
7 having a first set of entries, wherein the first set of entries is capable of being mapped to
8 a second set of entries of a second redirection table implemented in the network interface;
9 and

10 (ii) map the first set of entries to the second set of entries, based on the
11 number of conflicting entries in the first redirection table.

1 10. The system of claim 9, wherein the first redirection table is a software
2 redirection table, wherein the second redirection table is a hardware redirection table, and
3 wherein a conflict is caused if at least two entries of the software redirection table that are
4 capable of being mapped to one entry of the hardware redirection table indicate different
5 receive queues, wherein the program logic is further capable of causing the at least one
6 processor to be operable to:

7 determine whether the first set of entries in the software redirection table has
8 more members than the second set of entries in the hardware redirection table, wherein
9 the number of conflicting entries are determined in response to a determination that the
10 first set of entries in the software redirection table has more members than the second set
11 of entries in the hardware redirection table; and

12 indicate that packets associated with conflicting entries are to be directed to one
13 receive queue, if the number of conflicting entries is less than a threshold.

1 11. The system of claim 10, wherein the program logic is further capable of
2 causing the at least one processor to be operable to:

3 distribute packets in the one receive queue among all processors for processing;
4 and

5 process packets in other receive queues in different processors.

1 12. The system of claim 10, wherein the program logic is further capable of
2 causing the at least one processor to be operable to:

3 indicate that all packets are to be directed to a single receive queue, if the number
4 of conflicting entries is not less than the threshold

1 13. The system of claim 12, further comprising:
2 a device driver, wherein the device driver is operable to process receive side
3 scaling in software by creation of virtual queues, and wherein the device driver is capable
4 of queuing deferred procedure calls associated with the virtual queues to corresponding
5 processors.

1 14. The system of claim 10, wherein the program logic is further capable of
2 causing the at least one processor to be operable to:
3 program the hardware redirection table in accordance with the software
4 redirection table, in response to the determination that the first set of entries in the
5 software redirection table does not have more members than the second set of entries in
6 the hardware redirection table.

1 15. The system of claim 9, further comprising:
2 a device driver operable to determine the number of conflicting entries and map
3 the first set of entries.

1 16. The system of claim 9, wherein the first redirection table is associated
2 with an operating system that supports receive side scaling, wherein the second
3 redirection table is implemented in the network interface, and wherein the second
4 redirection table is of a fixed size.

1 17. A system, comprising:
2 a computational platform;
3 a storage controller implemented in the computational platform;
4 at least one processor coupled to the computational platform;

5 a network interface coupled to computational platform; and
6 program logic including code that is capable of causing the at least one processor
7 to be operable to:

8 (i) determine a number of conflicting entries in a first redirection table
9 having a first set of entries, wherein the first set of entries is capable of being mapped to
10 a second set of entries of a second redirection table, wherein the second redirection table
11 is implemented in the network interface; and
12 (ii) map the first set of entries to the second set of entries, based on the
13 number of conflicting entries in the first redirection table.

1 18. The system of claim 17, wherein the first redirection table is a software
2 redirection table, wherein the second redirection table is a hardware redirection table, and
3 wherein a conflict is caused if at least two entries of the software redirection table that are
4 capable of being mapped to one entry of the hardware redirection table indicate different
5 receive queues, wherein the program logic is further capable of causing the at least one
6 processor to be operable to:

7 determine whether the first set of entries in the software redirection table has
8 more members than the second set of entries in the hardware redirection table, wherein
9 the number of conflicting entries are determined in response to a determination that the
10 first set of entries in the software redirection table has more members than the second set
11 of entries in the hardware redirection table; and

12 indicate that packets associated with conflicting entries are to be directed to one
13 receive queue, if the number of conflicting entries is less than a threshold.

1 19. The system of claim 18, wherein the program logic is further capable of
2 causing the at least one processor to be operable to:

3 distribute packets in the one receive queue among all processors for processing;
4 and
5 process packets in other receive queues in different processors.

1 20. The system of claim 18, wherein the program logic is further capable of
2 causing the at least one processor to be operable to:

3 indicate that all packets are to be directed to a single receive queue, in response to
4 the determination that the number of conflicting entries is not less than the threshold.

1 21. An article of manufacture, comprising a storage medium having stored
2 therein instructions that are operable by a machine to:

3 determine a number of conflicting entries in a first redirection table having a first
4 set of entries, wherein the first set of entries is capable of being mapped to a second set of
5 entries of a second redirection table; and

6 map the first set of entries to the second set of entries, based on the number of
7 conflicting entries in the first redirection table

1 22. The article of manufacture of claim 21, wherein the first redirection table
2 is a software redirection table, wherein the second redirection table is a hardware
3 redirection table, and wherein a conflict is caused if at least two entries of the software
4 redirection table that are capable of being mapped to one entry of the hardware
5 redirection table indicate different receive queues, wherein the instructions are further
6 operable by a machine to:

7 determine whether the first set of entries in the software redirection table has
8 more members than the second set of entries in the hardware redirection table, wherein
9 the number of conflicting entries are determined in response to determining that the first
10 set of entries in the software redirection table has more members than the second set of
11 entries in the hardware redirection table; and

12 indicate that packets associated with conflicting entries are to be directed to one
13 receive queue, in response to determining that the number of conflicting entries is less
14 than a threshold.

1 23. The article of manufacture of claim 22, wherein the instructions are
2 further operable by a machine to:
3 distribute packets in the one receive queue among all processors for processing;
4 and
5 process packets in other receive queues in different processors.

1 24. The article of manufacture of claim 22, wherein the instructions are
2 further operable by a machine to:
3 indicate that all packets are to be directed to a single receive queue, in response to
4 determining that the number of conflicting entries is not less than the threshold.

1 25. The article of manufacture of claim 24, wherein the instructions are
2 further operable by a machine to:
3 process receive side scaling in by creation of virtual queues, wherein a device
4 driver is capable of queuing deferred procedure calls associated with the virtual queues to
5 corresponding processors.

1 26. The article of manufacture of claim 22, wherein the instructions are
2 further operable by a machine to:
3 program the hardware redirection table in accordance with the software
4 redirection table, in response to determining that the first set of entries in the software
5 redirection table does not have more members than the second set of entries in the
6 hardware redirection table.

1 27. The article of manufacture of claim 21, wherein determination of the
2 number of conflicting entries and mapping the first set of entries are performed by a
3 device driver in a computational platform having a plurality of processors.

1 28. The article of manufacture of claim 21, wherein the first redirection table
2 is associated with an operating system that supports receive side scaling, wherein the
3 second redirection table is implemented in the network interface, and wherein the second
4 redirection table is of a fixed size.